This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1-38 (canceled)

39. (Previously Presented)   A method that is computer implemented of evaluating a semijoin having a first operand, a second operand, and a predicate, said method comprising the acts of:

determining that said first and second operands each comprise first and second instances, respectively, of a common relation;

determining that said predicate is, or conjunctively includes, an equality comparison between one or more columns of said first instance of said relation and corresponding columns of said second instance of said relation;

segmenting said common relation based on said one or more columns to produce one or more segments of said common relation;

performing said semijoin separately on each of said segments;

processing a query in accordance with said semijoin, wherein said processing comprises transforming, by a computer processor, data in a memory according to the query to generate a query result; and

sending [[a]] the query result to an issuer of the query.

40. (Original) The method of claim 39, further comprising the act of spooling each of said segments in a memory location, wherein said performing act comprises applying said semijoin successively to the segments spooled in said memory location.

41. (Original) The method of claim 39, further comprising the act of compiling a SQL query which includes an EXISTS clause to produce a relational expression that includes said semijoin.

42. (Canceled)

43. (Previously Presented)   A method that is computer implemented of evaluating an anti-semijoin having a first operand, a second operand, and a predicate, said method comprising the acts of:

determining that said first and second operands each comprise first and second instances, respectively, of a common relation;

determining that said predicate is, or conjunctively includes, an equality comparison between one or more columns of said first instance of said relation and corresponding columns of said second instance of said relation;

segmenting said common relation based on said one or more columns to produce one or more segments of said common relation;

performing said anti-semijoin separately on each of said segments;

processing a query in accordance with said anti-semijoin, wherein said processing comprises transforming, by a computer processor, data in a memory according to the query to generate a query result; and

sending [[a]] the query result to an issuer of the query.

44. (Original) The method of claim 43, further comprising the act of spooling each of said segments in a memory location, wherein said performing act comprises applying said anti-semijoin successively to the segments spooled in said memory location.

45. (Original) The method of claim 43, further comprising the act of compiling a SQL query which includes a NOT EXISTS clause to produce a relational expression that includes said anti-semijoin.

46. (Canceled)

47. (Previously Presented)   A method that is computer implemented of sorting rows of a database table according to a first set of one or more columns, said rows having been sorted on a second set of one or more columns, said method comprising the acts of:

segmenting said database table based on said second set of columns to produce segments of said database table, each of said segments comprising rows having common

values in said second set of columns;

          separately sorting each of said segments based on the values in said first set of columns, wherein said segmenting act is performed without spooling rows of said segments;

          processing a query in accordance with the sorted segments, wherein said processing comprises transforming, by a computer processor, data in a memory according to the query to generate a query result; and

          sending [[a]] the query result to an issuer of the query.

48. (Canceled)

49. (Original) The method of claim 47, wherein said segmenting act comprises:

          identifying a first row of said database table, said first rows having first values in said second set of columns; and

          identifying a second row of said database table, said second row having said first values in said second set of columns, said database table having an order, said second row being the last occurring row in said order having said first values in said second set of columns.

50. (Original) The method of claim 47, wherein said act of separately sorting comprises:

          invoking a first function which sorts rows in first one of said segments;

          invoking a second function which identifies a next one of said segments; and

          repeating said acts of invoking said first and second function until all of said segments have been exhausted.

51. (Original) The method of claim 47, further comprising the act of:

          creating an expression tree including a GbApply operator having:

               a first child sub-tree specifying said database table as input to said GbApply operator;

               segmentation data indicating that said database table is to be segmented on said second set of columns; and

a second child sub-tree specifying a relational fragment to be performed on each of said segments, said relational fragment specifying a sort of the rows of said segments on said first set of columns.

52. (Canceled)

53. (Previously Presented)   In a database system that is computer implemented which performs operations on a table having rows and columns, a method of identifying a row having a superlative value in a first of said columns from among a set of rows having a common value in a second of said columns, said method comprising the acts of:

      segmenting said rows based on the values in said second column to produce groups of one or more rows, wherein all of the rows in a first of said groups have a common value in said second column;

      sorting the rows in said first group based on the values in said first column;

      identifying the first or last row in said first group;

      processing a query in accordance with the identified row, wherein said processing comprises transforming, by a computer processor, data in a memory according to the query to generate a query result; and

      sending [[a]] the query result to an issuer of the query.

54. (Original) The method of claim 53, wherein said superlative value comprises a maximum value, and wherein said identifying act comprises identifying the last row in said first group.

55. (Original) The method of claim 53, wherein said superlative value comprises a minimum value, and wherein said identifying act comprises identifying the first row in said first group.

56. (Original) The method of claim 53, wherein said segmenting act comprises:

      sorting said rows based on the values in said second column; and

following said sorting act, identifying the first and last rows which have said common value in said second column.

57. (Original) The method of claim 53, further comprising the act of:

creating an expression tree including a GbApply operator having:

a first child sub-tree specifying said table as input to said GbApply operator;

segmentation data indicating that said database table is to be segmented on said second set of columns; and

a second child sub-tree specifying a relational fragment to be performed on each of said segments, said relational fragment specifying the selection of either the first or last of the rows in a segment.

58-66. (Canceled)

67. (Previously Presented)   A computer-readable storage medium having computer executable instructions for performing the following acts:

determining that a relational expression includes a join operator having:

a first operand which includes a first instance of a relation based on information stored in a database, said relation having a set of rows and a set of columns;

a second operand including a second instance of said relation; and

a join predicate which is, or conjunctively includes, an equality comparison between one or more columns of said first instance of said relation and corresponding one or more columns of said second instance of said relation;

segmenting said relation based on said one or more columns to produce one or more segments of said relation;

applying said join operator separately to each of said segments;

processing a query in accordance with said join operator and said segments, wherein said processing comprises transforming, by a computer processor, data in a memory according to the query to generate a query result; and

sending [[a]] the query result to an issuer of the query.

68. (Previously Presented)   The computer-readable storage medium of claim 67, having further computer executable instructions for performing the act of identifying said relational expression as a fragment of a larger expression.

69. (Previously Presented)   The computer-readable storage medium of claim 67, having further computer executable instructions for performing the acts of:

        creating a first expression tree representative of said relational expression; and

        creating a second expression tree based on said first expression tree, said second expression tree including a node having:

                data indicative of said relation;

                data indicative of said one or more columns; and

                data indicative of said join operator.

70. (Previously Presented)   The computer-readable storage medium of claim 69, having further computer executable instructions for performing the act of determining that second expression can be evaluated more efficiently than said first expression tree.

71. (Previously Presented)   The computer-readable storage medium of claim 67, wherein said join operator comprises one of: inner join, semijoin, or anti-semijoin.

72. (Previously Presented)   The computer-readable storage medium of claim 67, having further computer executable instructions for performing the act of determining that said first operand further includes a filter which modifies said first instance of said relation according to a second predicate.

73. (Previously Presented)   The computer-readable storage medium of claim 67, having further computer executable instructions for performing the act of determining that said first operand further includes an aggregate operation which specifies the computation of a value based on one or more rows of said relation.

74. (Previously Presented)   The computer-readable storage medium of claim 67, having further computer executable instructions for performing the act of successively spooling each of said segments, wherein said applying act comprises applying said join operator to the segment in said spool.

75. (Previously Presented)   The computer-readable storage medium of claim 67, wherein said relation comprises a stored table.

76. (Previously Presented)   The computer-readable storage medium of claim 67, wherein said relation comprises a sub-expression based on one or more stored tables, and wherein said computer-readable medium further comprises the act of evaluating said sub-expression to produce said relation.

77. (Previously Presented)   The computer-readable storage medium of claim 67, having further computer executable instructions for performing the act of compiling a query to produce said relational expression.

78. (Previously Presented)   The computer-readable storage medium of claim 67, wherein said segmenting act comprises:

        sorting said relation;

        spooling a first row of the sorted relation; and

        spooling successive rows of the sorted relation until a row is encountered having values in said one or more columns that differ from the values in said one or more columns of said first row.

79. (Previously Presented)   A computer-readable storage medium having computer executable instructions for performing the following acts:

        determining that a relational expression includes:

            a first join operator having:

a first operand which includes a first instance of a first relation based on information stored in a database, said first relation having a set of rows and a set of columns;

a second operand including a second instance of said first relation; and

a first predicate which is, or conjunctively includes, an equality comparison between one or more columns of said first instance of said first relation and corresponding one or more columns of said second instance of said first relation; and

a second join operator having:

a third operand which includes the result of said first join operator;

a fourth operand including a second relation based on information stored in said database; and

a second predicate;

performing a join of said second relation and said first relation using said second predicate to produce an intermediate result;

segmenting said intermediate result based on said one or more columns to produce one or more segments of said intermediate result;

applying said first join operator including said first predicate separately to each of said segments;

processing a query in accordance with said first join operator, said first predicate and said segments, wherein said processing comprises transforming, by a computer processor, data in a memory according to the query to generate a query result; and

sending [[a]] the query result to an issuer of the query.

80. (Previously Presented)   The computer-readable storage medium of claim 79, wherein said segmenting act is further based on a key of said second relation.

81. (Previously Presented)   The computer-readable storage medium of claim 79, wherein said segmenting act is further based on all columns of said second relation.

82. (Previously Presented)   The computer-readable storage medium of claim 79, having further computer executable instructions for performing the act of:

determining that said second predicate references only columns in the set of columns consisting of:

said one or more columns; and

columns in said second relation.

83. (Previously Presented)   The computer-readable storage medium of claim 79, having further computer executable instructions for performing the acts of:

removing conjuncts from said second predicate that reference columns other than:

said one or more columns; and

columns in said second relation; and

removing rows from the result of said applying step that do not satisfy the removed conjuncts.

84. (Previously Presented)   The computer-readable storage medium of claim 79, having further computer executable instructions for performing the act of comparing the predicted efficiency of at least two procedures of evaluating said relational expression.

85. (Previously Presented)   The computer-readable storage medium of claim 79, having further computer executable instructions for performing the act of identifying said relational expression as a fragment of a larger expression.

86. (Previously Presented)   The computer-readable storage medium of claim 79, wherein either said first join operator or said second join operator comprises one of: inner join, semijoin, or anti-semijoin.

87. (Previously Presented)   The computer-readable storage medium of claim 79, having further computer executable instructions for performing the act of determining that

said first operand further includes a filter which modifies said first instance of said first

relation according to a third predicate.

88. (Previously Presented)   The computer-readable storage medium of claim 79,

having further computer executable instructions for performing the act of determining that

said first operand further includes an aggregate operation which specifies the computation of

a value based on one or more rows of said first relation.

89. (Previously Presented)   The computer-readable storage medium of claim 79,

having further computer executable instructions for performing the act of successively

spooling each of said segments, wherein said applying act comprises applying said first join

operator to the segment in said spool.

90. (Previously Presented)   The computer-readable storage medium of claim 79,

wherein said first relation comprises a stored table.

91. (Previously Presented)   The computer-readable storage medium of claim 79,

wherein said first relation comprises a sub-expression based on one or more stored tables, and

wherein said computer-readable medium further comprises the act of evaluating said sub-

expression to produce said first relation.

92. (Previously Presented)   The computer-readable storage medium of claim 79,

wherein said segmenting act comprises:

       sorting said intermediate result;

       spooling a first row of the sorted intermediate result; and

       spooling successive rows of the sorted intermediate until a row is encountered

having values in said one or more columns that differ from the values in said one or more

columns of said first row.

93. (Previously Presented)   A computer-readable storage medium of providing

information from a database comprising the acts of:

determining that a first relational expression includes a join operator having:

a first operand which includes a first relation based on information stored in said database;

a second operand which includes an operator calling for a relational fragment to be applied successively to segments of said second relation, said second relation being based on information stored in said database, said segments of said second relation being based on distinct values of one or more columns of said second relation; and

a predicate;

creating a second relational expression that includes:

an operator calling for said relational fragment to be applied successively to segments of a third relation, said segments of said third relation being based on distinct values of said one or more columns and all columns of said first relation, said third relation being the result of a join operator having:

a third operand which includes said first relation;

a fourth operand which includes said second relation; and

said predicate;

evaluating said second relational expression;

processing a query in accordance with the evaluated second relational expression, wherein said processing comprises transforming, by a computer processor, data in a memory according to the query to generate a query result; and

sending [[a]] the query result to an issuer of the query.

94. (Previously Presented)   The computer-readable storage medium of claim 93, having further computer executable instructions for performing the act of determining that said predicate references only columns in the set of columns consisting of:

said one or more columns; and

columns in said first relation.

95. (Previously Presented)   The computer-readable storage medium of claim 93, having further computer executable instructions for performing the acts of:

removing conjuncts from said predicate that reference columns other than:

said one or more columns; and

columns in said first relation; and

removing rows from the result of said evaluating step that do not satisfy the

removed conjuncts.

96. (Previously Presented)   The computer-readable storage medium of claim 93,

wherein said join operator comprises one of: inner join, semijoin, or anti-semijoin.